

Script generated by TTT

Title: Seidl: Theoretische_Informatik
(06.05.2013)

Date: Mon May 06 10:21:26 CEST 2013

Duration: 86:48 min

Pages: 75

2.11 Minimierung endlicher Automaten

- ① Beispiele
- ② Algorithmen
- ③ Minimalitätsbeweis

Der Algorithmus zur Minimierung eines DFA:

Der Algorithmus zur Minimierung eines DFA:

- ① Entferne alle von q_0 aus nicht erreichbaren Zustände.
- ② *Berechne die äquivalenten Zustände des Automaten.*

Der Algorithmus zur Minimierung eines DFA:

- 1 Entferne alle von q_0 aus nicht erreichbaren Zustände.
- 2 Berechne die *äquivalenten* Zustände des Automaten.

Der Algorithmus zur Minimierung eines DFA:

- 1 Entferne alle von q_0 aus nicht erreichbaren Zustände.
- 2 Berechne die *äquivalenten* Zustände des Automaten.
- 3 Kollabiere den Automaten durch Zusammenfassung aller äquivalenten Zustände.

Zustände p und q sind *unterscheidbar* wenn es $w \in \Sigma^*$ gibt mit $\hat{\delta}(p, w) \in F$ und $\hat{\delta}(q, w) \notin F$ oder umgekehrt.

Zustände sind *äquivalent* wenn sie nicht unterscheidbar sind, d.h. wenn für alle $w \in \Sigma^*$ gilt:

$$\hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F$$

Der Algorithmus zur Minimierung eines DFA:

- 1 Entferne alle von q_0 aus nicht erreichbaren Zustände.
- 2 Berechne die *äquivalenten* Zustände des Automaten.
- 3 Kollabiere den Automaten durch Zusammenfassung aller äquivalenten Zustände.

Zustände p und q sind *unterscheidbar* wenn es $w \in \Sigma^*$ gibt mit $\hat{\delta}(p, w) \in F$ und $\hat{\delta}(q, w) \notin F$ oder umgekehrt.

Zustände sind *äquivalent* wenn sie nicht unterscheidbar sind, d.h. wenn für alle $w \in \Sigma^*$ gilt:

$$\hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F$$

Sind $\delta(p, a)$ und $\delta(q, a)$ unterscheidbar, dann auch p und q .

Der Algorithmus zur Minimierung eines DFA:

- 1 Entferne alle von q_0 aus nicht erreichbaren Zustände.
- 2 Berechne die *äquivalenten* Zustände des Automaten.
- 3 Kollabiere den Automaten durch Zusammenfassung aller äquivalenten Zustände.

Zustände p und q sind *unterscheidbar* wenn es $w \in \Sigma^*$ gibt mit $\hat{\delta}(p, w) \in F$ und $\hat{\delta}(q, w) \notin F$ oder umgekehrt.

Zustände sind *äquivalent* wenn sie nicht unterscheidbar sind, d.h. wenn für alle $w \in \Sigma^*$ gilt:

$$\hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F$$

Sind $\delta(p, a)$ und $\delta(q, a)$ unterscheidbar, dann auch p und q .

\Rightarrow Unterscheidbarkeit pflanzt sich rückwärts fort.

Berechnung äquivalenter Zustände eines DFA



Berechnung äquivalenter Zustände eines DFA

durch Berechnung der unterscheidbaren Zustände

Eingabe: DFA $A = (Q, \Sigma, \delta, q_0, F)$

Ausgabe: Äquivalenzrelation auf Q .

Datenstruktur: Eine Menge U ungeordneter Paare $\{p, q\} \subseteq Q$.



Berechnung äquivalenter Zustände eines DFA

durch Berechnung der unterscheidbaren Zustände

Eingabe: DFA $A = (Q, \Sigma, \delta, q_0, F)$

Ausgabe: Äquivalenzrelation auf Q .

Datenstruktur: Eine Menge U ungeordneter Paare $\{p, q\} \subseteq Q$.

Algorithmus U:

1 $U := \{\{p, q\} \mid p \in F \wedge q \notin F\}$

Berechnung äquivalenter Zustände eines DFA

durch Berechnung der unterscheidbaren Zustände

Eingabe: DFA $A = (Q, \Sigma, \delta, q_0, F)$

Ausgabe: Äquivalenzrelation auf Q .

Datenstruktur: Eine Menge U ungeordneter Paare $\{p, q\} \subseteq Q$.

Algorithmus U:

1 $U := \{\{p, q\} \mid p \in F \wedge q \notin F\}$

2 **while** $\exists \{p, q\} \notin U. \exists a \in \Sigma. \{\delta(p, a), \delta(q, a)\} \in U$
do $U := U \cup \{\{p, q\}\}$

Berechnung äquivalenter Zustände eines DFA

durch Berechnung der unterscheidbaren Zustände

Eingabe: DFA $A = (Q, \Sigma, \delta, q_0, F)$

Ausgabe: Äquivalenzrelation auf Q .

Datenstruktur: Eine Menge U ungeordneter Paare $\{p, q\} \subseteq Q$.

Algorithmus U:

- 1 $U := \{\{p, q\} \mid p \in F \wedge q \notin F\}$
- 2 **while** $\exists \{p, q\} \notin U. \exists a \in \Sigma. \{\delta(p, a), \delta(q, a)\} \in U$
do $U := U \cup \{\{p, q\}\}$

Invariante: $\{p, q\} \in U \implies p$ und q unterscheidbar

Berechnung äquivalenter Zustände eines DFA

durch Berechnung der unterscheidbaren Zustände

Eingabe: DFA $A = (Q, \Sigma, \delta, q_0, F)$

Ausgabe: Äquivalenzrelation auf Q .

Datenstruktur: Eine Menge U ungeordneter Paare $\{p, q\} \subseteq Q$.

Algorithmus U:

- 1 $U := \{\{p, q\} \mid p \in F \wedge q \notin F\}$
- 2 **while** $\exists \{p, q\} \notin U. \exists a \in \Sigma. \{\delta(p, a), \delta(q, a)\} \in U$
do $U := U \cup \{\{p, q\}\}$

Invariante: $\{p, q\} \in U \implies p$ und q unterscheidbar

Lemma 2.50

Am Ende gilt: U ist Menge aller unterscheidbaren Zustände.

! spare

Berechnung äquivalenter Zustände eines DFA

durch Berechnung der unterscheidbaren Zustände

Eingabe: DFA $A = (Q, \Sigma, \delta, q_0, F)$

Ausgabe: Äquivalenzrelation auf Q .

Datenstruktur: Eine Menge U ungeordneter Paare $\{p, q\} \subseteq Q$.

Algorithmus U:

- 1 $U := \{\{p, q\} \mid p \in F \wedge q \notin F\}$
- 2 **while** $\exists \{p, q\} \notin U. \exists a \in \Sigma. \{\delta(p, a), \delta(q, a)\} \in U$
do $U := U \cup \{\{p, q\}\}$

Invariante: $\{p, q\} \in U \implies p$ und q unterscheidbar

Lemma 2.50

Am Ende gilt: U ist Menge aller unterscheidbaren Zustände.

Beweis:

$\{p, q\} \in U \implies p$ und q unterscheidbar: Invariante

p und q unterscheidbar $\implies \{p, q\} \in U$:

! spare

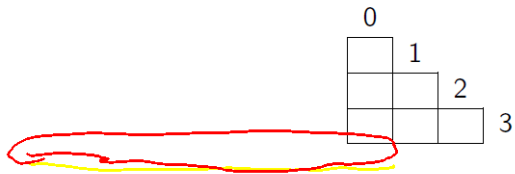
Implementierung von U :

Tabelle von anfangs unmarkierten Paaren $\{p, q\}, p \neq q$.

! spare

Implementierung von U :

Tabelle von anfangs unmarkierten Paaren $\{p, q\}, p \neq q$.



! spare

Implementierung von U :

Tabelle von anfangs unmarkierten Paaren $\{p, q\}, p \neq q$.

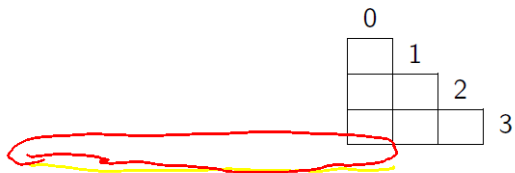


for all $p \in F, q \in Q \setminus F$ **do** markiere $\{p, q\}$

! spare

Implementierung von U :

Tabelle von anfangs unmarkierten Paaren $\{p, q\}, p \neq q$.



for all $p \in F, q \in Q \setminus F$ **do** markiere $\{p, q\}$
while \exists unmarkiertes $\{p, q\}$

! spare

Implementierung von U :

Tabelle von anfangs unmarkierten Paaren $\{p, q\}, p \neq q$.

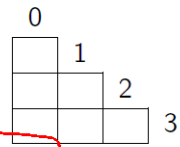


for all $p \in F, q \in Q \setminus F$ **do** markiere $\{p, q\}$
while \exists unmarkiertes $\{p, q\} \exists a \in \Sigma. \{\delta(p, a), \delta(q, a)\}$ ist markiert

! spare

Implementierung von U :

Tabelle von anfangs unmarkierten Paaren $\{p, q\}, p \neq q$.



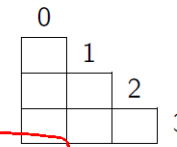
for all $p \in F, q \in Q \setminus F$ **do** markiere $\{p, q\}$
while \exists unmarkiertes $\{p, q\} \exists a \in \Sigma. \{\delta(p, a), \delta(q, a)\}$ ist markiert
do markiere $\{p, q\}$

Komplexität:

! spare

Implementierung von U :

Tabelle von anfangs unmarkierten Paaren $\{p, q\}, p \neq q$.



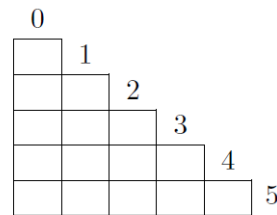
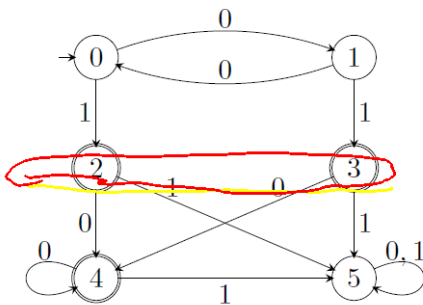
for all $p \in F, q \in Q \setminus F$ **do** markiere $\{p, q\}$
while \exists unmarkiertes $\{p, q\} \exists a \in \Sigma. \{\delta(p, a), \delta(q, a)\}$ ist markiert
do markiere $\{p, q\}$

Komplexität:

$$O\left(\binom{n}{2} + \binom{n}{2} \binom{n}{2} |\Sigma|\right)$$

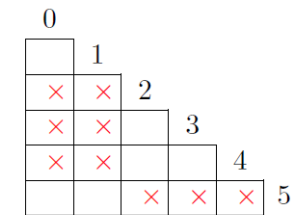
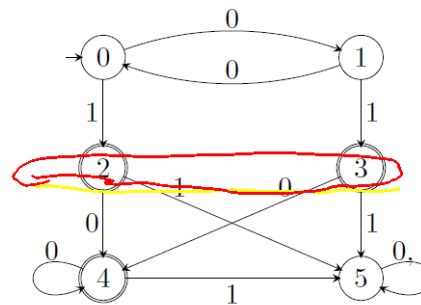
! spare

Beispiel 2.51



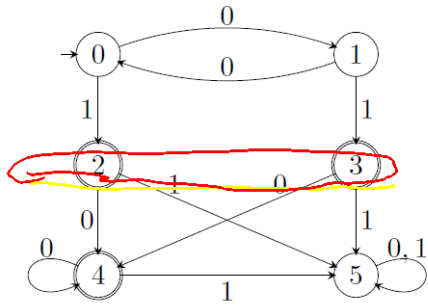
! spare

Beispiel 2.51



! spare

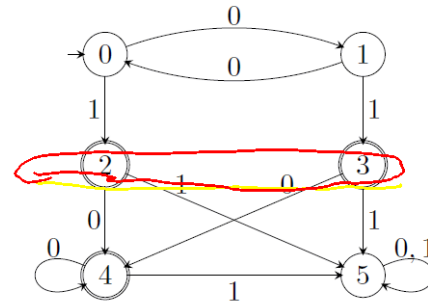
Beispiel 2.51



	1					
x	x	2				
x	x		3			
x	x			4		
x	x	x	x	x	5	

5 spare

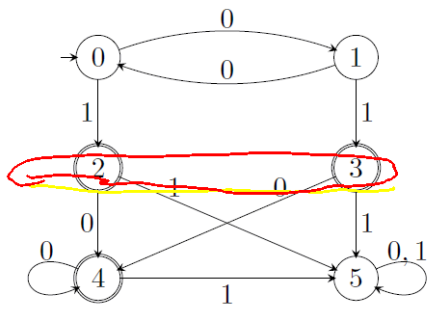
Beispiel 2.51



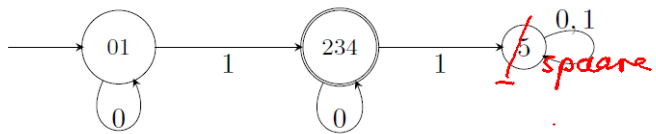
	1					
x	x	2				
x	x		3			
x	x			4		
x	x	x	x	x	5	

5 spare

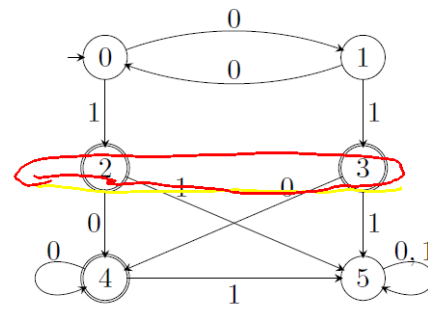
Beispiel 2.51



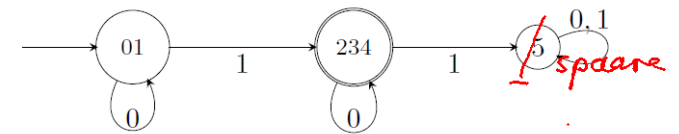
	1					
x	x	2				
x	x		3			
x	x			4		
x	x	x	x	x	5	



Beispiel 2.51



	1					
x	x	2				
x	x		3			
x	x			4		
x	x	x	x	x	5	



Von $O(n^4)$ zu $O(n^2)$ mit Abhängigkeitsanalyse:

$\{p', q'\}$ unterscheidbar \implies
 $\{p, q\}$ unterscheidbar falls $\{p, q\} \xrightarrow{a} \{p', q'\}$

$D[\{p', q'\}]$: Menge der Paare $\{p, q\}$ wie oben, anfangs leer



! spare

Von $O(n^4)$ zu $O(n^2)$ mit Abhängigkeitsanalyse:

$\{p', q'\}$ unterscheidbar \implies
 $\{p, q\}$ unterscheidbar falls $\{p, q\} \xrightarrow{a} \{p', q'\}$

$D[\{p', q'\}]$: Menge der Paare $\{p, q\}$ wie oben, anfangs leer

for all $\{p, q\} \subseteq Q$ mit $p \neq q$,



! spare

Von $O(n^4)$ zu $O(n^2)$ mit Abhängigkeitsanalyse:

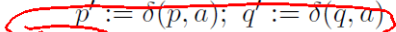
$\{p', q'\}$ unterscheidbar \implies
 $\{p, q\}$ unterscheidbar falls $\{p, q\} \xrightarrow{a} \{p', q'\}$

$D[\{p', q'\}]$: Menge der Paare $\{p, q\}$ wie oben, anfangs leer

for all $\{p, q\} \subseteq Q$ mit $p \neq q$, $a \in \Sigma$ **do**

$p' := \delta(p, a)$; $q' := \delta(q, a)$

if $p' \neq q'$ **then** $D[\{p, q\}] := D[\{p', q'\}] \cup \{\{p, q\}\}$



! spare

Von $O(n^4)$ zu $O(n^2)$ mit Abhängigkeitsanalyse:

$\{p', q'\}$ unterscheidbar \implies
 $\{p, q\}$ unterscheidbar falls $\{p, q\} \xrightarrow{a} \{p', q'\}$

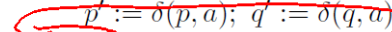
$D[\{p', q'\}]$: Menge der Paare $\{p, q\}$ wie oben, anfangs leer

for all $\{p, q\} \subseteq Q$ mit $p \neq q$, $a \in \Sigma$ **do**

$p' := \delta(p, a)$; $q' := \delta(q, a)$

if $p' \neq q'$ **then** $D[\{p, q\}] := D[\{p', q'\}] \cup \{\{p, q\}\}$

for all $p' \in F$, $q' \in Q \setminus F$ **do** $mark(\{p', q'\})$



! spare

Von $O(n^4)$ zu $O(n^2)$ mit Abhängigkeitsanalyse:

$\{p', q'\}$ unterscheidbar \implies
 $\{p, q\}$ unterscheidbar falls $\{p, q\} \xrightarrow{a} \{p', q'\}$

$D[\{p', q'\}]$: Menge der Paare $\{p, q\}$ wie oben, anfangs leer

for all $\{p, q\} \subseteq Q$ mit $p \neq q, a \in \Sigma$ **do**

$p' := \delta(p, a); q' := \delta(q, a)$

if $p' \neq q'$ **then** $D[\{p', q'\}] := D[\{p', q'\}] \cup \{\{p, q\}\}$

for all $p' \in F, q' \in Q \setminus F$ **do** $mark(\{p', q'\})$

$mark(\{p', q'\}) =$

if $\{p', q'\}$ unmarkiert **then**

markiere $\{p', q'\}$

for all $p, q \in D[\{p', q'\}]$ **do** $mark(pq)$

! spare

Von $O(n^4)$ zu $O(n^2)$ mit Abhängigkeitsanalyse:

$\{p', q'\}$ unterscheidbar \implies
 $\{p, q\}$ unterscheidbar falls $\{p, q\} \xrightarrow{a} \{p', q'\}$

$D[\{p', q'\}]$: Menge der Paare $\{p, q\}$ wie oben, anfangs leer

for all $\{p, q\} \subseteq Q$ mit $p \neq q, a \in \Sigma$ **do**

$p' := \delta(p, a); q' := \delta(q, a)$

if $p' \neq q'$ **then** $D[\{p', q'\}] := D[\{p', q'\}] \cup \{\{p, q\}\}$

for all $p' \in F, q' \in Q \setminus F$ **do** $mark(\{p', q'\})$

$mark(\{p', q'\}) =$

if $\{p', q'\}$ unmarkiert **then**

markiere $\{p', q'\}$

for all $p, q \in D[\{p', q'\}]$ **do** $mark(pq)$

! spare

Komplexität: $O(n^2 + n^2) = O(n^2)$.



John Hopcroft

~~An $n \log n$ Algorithm for Minimizing the States in a Finite Automaton.~~ 1971.

! spare

Noch eine Anwendung: Äquivalenztest von DFAs.

! spare

Noch eine Anwendung: Äquivalenztest von DFAs.

- 1 Gegeben DFAs A und B , bilde disjunkte Vereinigung.
(„Male A und B nebeneinander.“)

! spare

Noch eine Anwendung: Äquivalenztest von DFAs.

- 1 Gegeben DFAs A und B , bilde disjunkte Vereinigung.
(„Male A und B nebeneinander.“)
- 2 Berechne Menge der äquivalenten Zustände.
- 3 $L(A) = L(B)$ gdw die beiden Startzustände äquivalent sind.

! spare

Bisher: Der Minimierungsalgorithmus (zur Erinnerung).

- 1 Entferne alle von q_0 aus nicht erreichbaren Zustände.
- 2 Berechne die *äquivalenten* Zustände des Automaten.
- 3 Kollabiere den Automaten durch Zusammenfassung aller äquivalenten Zustände.

! spare

Bisher: Der Minimierungsalgorithmus (zur Erinnerung).

- 1 Entferne alle von q_0 aus nicht erreichbaren Zustände.
- 2 Berechne die *äquivalenten* Zustände des Automaten.
- 3 Kollabiere den Automaten durch Zusammenfassung aller äquivalenten Zustände.

Jetzt: Die Präzisierung.

- 1 Was ist der „kollabierete Automat“?
- 2 Ist das wirklich der minimale Automat?

! spare

Eine Relation $\approx \subseteq A \times A$ ist eine **Äquivalenzrelation** falls

- Reflexivität: $\forall a \in A. a \approx a$
- Symmetrie: $\forall a, b \in A. a \approx b \implies b \approx a$
- Transitivität: $\forall a, b, c \in A. a \approx b \wedge b \approx c \implies a \approx c$

$[a]_{\approx}$

! spare

Eine Relation $\approx \subseteq A \times A$ ist eine **Äquivalenzrelation** falls

- Reflexivität: $\forall a \in A. a \approx a$
- Symmetrie: $\forall a, b \in A. a \approx b \implies b \approx a$
- Transitivität: $\forall a, b, c \in A. a \approx b \wedge b \approx c \implies a \approx c$

Äquivalenzklasse:

$[a]_{\approx} := \{b \mid a \approx b\}$

! spare

Eine Relation $\approx \subseteq A \times A$ ist eine **Äquivalenzrelation** falls

- Reflexivität: $\forall a \in A. a \approx a$
- Symmetrie: $\forall a, b \in A. a \approx b \implies b \approx a$
- Transitivität: $\forall a, b, c \in A. a \approx b \wedge b \approx c \implies a \approx c$

Äquivalenzklasse:

$[a]_{\approx} := \{b \mid a \approx b\}$

Es gilt:

$$[a]_{\approx} = [b]_{\approx} \iff a \approx b$$

! spare

Eine Relation $\approx \subseteq A \times A$ ist eine **Äquivalenzrelation** falls

- Reflexivität: $\forall a \in A. a \approx a$
- Symmetrie: $\forall a, b \in A. a \approx b \implies b \approx a$
- Transitivität: $\forall a, b, c \in A. a \approx b \wedge b \approx c \implies a \approx c$

Äquivalenzklasse:

$[a]_{\approx} := \{b \mid a \approx b\}$

Es gilt:

$$[a]_{\approx} = [b]_{\approx} \iff a \approx b$$

Quotientenmenge:

$$A/\approx := \{[a]_{\approx} \mid a \in A\}$$

! spare

Im Folgenden sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DFA ohne unerreichbare Zustände.



! spare

Im Folgenden sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DFA ohne unerreichbare Zustände.

Definition 2.52 (Äquivalenz von Zuständen)

$$p \equiv_A q \Leftrightarrow (\forall w \in \Sigma^*. \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F)$$



! spare

Im Folgenden sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DFA ohne unerreichbare Zustände.

Definition 2.52 (Äquivalenz von Zuständen)

$$p \equiv_A q \Leftrightarrow (\forall w \in \Sigma^*. \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F)$$

Fakt 2.53

\equiv_A ist eine Äquivalenzrelation.



! spare

Im Folgenden sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DFA ohne unerreichbare Zustände.

Definition 2.52 (Äquivalenz von Zuständen)

$$p \equiv_A q \Leftrightarrow (\forall w \in \Sigma^*. \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F)$$

Fakt 2.53

\equiv_A ist eine Äquivalenzrelation.



Wir schreiben \equiv statt \equiv_A wenn A klar ist.

! spare

Im Folgenden sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DFA ohne unerreichbare Zustände.

Definition 2.52 (Äquivalenz von Zuständen)

$$p \equiv_A q \Leftrightarrow (\forall w \in \Sigma^*. \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F)$$

Fakt 2.53

\equiv_A ist eine Äquivalenzrelation.

Wir schreiben \equiv statt \equiv_A wenn A klar ist.

Erinnerung:

Lemma 2.54

$$p \equiv_A q \implies \delta(p, a) \equiv_A \delta(q, a)$$

Kongruenz | sparsare

Im Folgenden sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DFA ohne unerreichbare Zustände.

Definition 2.52 (Äquivalenz von Zuständen)

$$p \equiv_A q \Leftrightarrow (\forall w \in \Sigma^*. \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F)$$

Fakt 2.53

\equiv_A ist eine Äquivalenzrelation.

Im Folgenden sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DFA ohne unerreichbare Zustände.

Definition 2.52 (Äquivalenz von Zuständen)

$$p \equiv_A q \Leftrightarrow (\forall w \in \Sigma^*. \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F)$$

Fakt 2.53

\equiv_A ist eine Äquivalenzrelation.

Wir schreiben \equiv statt \equiv_A wenn A klar ist.

Erinnerung:

Lemma 2.54

$$p \equiv_A q \implies \delta(p, a) \equiv_A \delta(q, a)$$

Kongruenz

Im Folgenden sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DFA ohne unerreichbare Zustände.

Definition 2.52 (Äquivalenz von Zuständen)

$$p \equiv_A q \Leftrightarrow (\forall w \in \Sigma^*. \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F)$$

Fakt 2.53

\equiv_A ist eine Äquivalenzrelation.

Wir schreiben \equiv statt \equiv_A wenn A klar ist.

Erinnerung:

Lemma 2.54

$$p \equiv_A q \implies \delta(p, a) \equiv_A \delta(q, a)$$

Lemma 2.55

Algorithmus U liefert die unterscheidbaren Zustände, also \neq .

Kongruenz

Im Folgenden sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DFA ohne unerreichbare Zustände.

Definition 2.52 (Äquivalenz von Zuständen)

$$p \equiv_A q \Leftrightarrow (\forall w \in \Sigma^*. \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F)$$

Fakt 2.53

\equiv_A ist eine Äquivalenzrelation.

Wir schreiben \equiv statt \equiv_A wenn A klar ist.

Erinnerung:

Lemma 2.54

$$p \equiv_A q \implies \delta(p, a) \equiv_A \delta(q, a)$$

Lemma 2.55

Algorithmus U liefert die unterscheidbaren Zustände, also \neq .

In der weiteren Analyse beziehen wir uns direkt auf \equiv , nicht mehr auf den Algorithmus.

Kongruenz

Die „Kollabierung“ von A bzgl. \equiv ist der Quotientenautomat:

Definition 2.56 (Quotientenautomat)

$$A/\equiv :=$$

Kongruenz

Die „Kollabierung“ von A bzgl. \equiv ist der Quotientenautomat:

Definition 2.56 (Quotientenautomat)

$$A/\equiv := (Q/\equiv, \Sigma, \delta', [q_0]_{\equiv}, F/\equiv)$$

Kongruenz

Die „Kollabierung“ von A bzgl. \equiv ist der Quotientenautomat:

Definition 2.56 (Quotientenautomat)

$$A/\equiv := (Q/\equiv, \Sigma, \delta', [q_0]_{\equiv}, F/\equiv)$$

Kongruenz

Die „Kollabierung“ von A bzgl. \equiv ist der *Quotientenautomat*:

Definition 2.56 (Quotientenautomat)

$$A/\equiv := (Q/\equiv, \Sigma, \delta', [q_0]_{\equiv}, F/\equiv)$$
$$\delta'([p]_{\equiv}, a) := \underline{[\delta(p, a)]_{\equiv}}$$

Kongruenz

Die „Kollabierung“ von A bzgl. \equiv ist der *Quotientenautomat*:

Definition 2.56 (Quotientenautomat)

$$A/\equiv := (Q/\equiv, \Sigma, \delta', [q_0]_{\equiv}, F/\equiv)$$
$$\delta'([p]_{\equiv}, a) := \underline{[\delta(p, a)]_{\equiv}}$$

Die Definition von δ' ist wohlgeformt da unabhängig von der Wahl des Repräsentanten p :

$$[p]_{\equiv}$$

Kongruenz

Die „Kollabierung“ von A bzgl. \equiv ist der *Quotientenautomat*:

Definition 2.56 (Quotientenautomat)

$$A/\equiv := (Q/\equiv, \Sigma, \delta', [q_0]_{\equiv}, F/\equiv)$$
$$\delta'([p]_{\equiv}, a) := \underline{[\delta(p, a)]_{\equiv}}$$

Die Definition von δ' ist wohlgeformt da unabhängig von der Wahl des Repräsentanten p :

$$[p]_{\equiv} = [p']_{\equiv} \implies p \equiv p' \implies \delta(p, a) \equiv \delta(p', a)$$

Kongruenz

Die „Kollabierung“ von A bzgl. \equiv ist der *Quotientenautomat*:

Definition 2.56 (Quotientenautomat)

$$A/\equiv := (Q/\equiv, \Sigma, \delta', [q_0]_{\equiv}, F/\equiv)$$
$$\delta'([p]_{\equiv}, a) := \underline{[\delta(p, a)]_{\equiv}}$$

Die Definition von δ' ist wohlgeformt da unabhängig von der Wahl des Repräsentanten p :

$$[p]_{\equiv} = [p']_{\equiv} \implies p \equiv p' \implies \delta(p, a) \equiv \delta(p', a)$$
$$\implies [\delta(p, a)]_{\equiv} = [\delta(p', a)]_{\equiv}$$

Lemma 2.57

$$L(A/\equiv) = L(A) \text{ Kongruenz}$$

Die „Kollabierung“ von A bzgl. \equiv ist der *Quotientenautomat*:

Definition 2.56 (Quotientenautomat)

$$A/\equiv := (Q/\equiv, \Sigma, \delta', [q_0]_{\equiv}, F/\equiv)$$
$$\delta'([p]_{\equiv}, a) := [\delta(p, a)]_{\equiv}$$

Die Definition von δ' ist wohlgeformt da unabhängig von der Wahl des Repräsentanten p :

$$[p]_{\equiv} = [p']_{\equiv} \implies p \equiv p' \implies \delta(p, a) \equiv \delta(p', a)$$
$$\implies [\delta(p, a)]_{\equiv} = [\delta(p', a)]_{\equiv}$$

Lemma 2.57

$$L(A/\equiv) = L(A) \text{ Kongruenz}$$

Beweis zur Übung.

Beobachtung

Für $p := \hat{\delta}(q_0, u)$ und $q := \hat{\delta}(q_0, v)$ gilt:

-

Kongruenz

Beobachtung

Für $p := \hat{\delta}(q_0, u)$ und $q := \hat{\delta}(q_0, v)$ gilt:

$$p \equiv_A q \Leftrightarrow \forall w \in \Sigma^*. \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F$$

-

Kongruenz

Beobachtung

Für $p := \hat{\delta}(q_0, u)$ und $q := \hat{\delta}(q_0, v)$ gilt:

$$p \equiv_A q \Leftrightarrow \forall w \in \Sigma^*. \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F$$
$$\Leftrightarrow \forall w \in \Sigma^*. \hat{\delta}(q_0, uw) \in F \Leftrightarrow \hat{\delta}(q_0, vw) \in F$$

-

Kongruenz

Beobachtung

Für $p := \hat{\delta}(q_0, u)$ und $q := \hat{\delta}(q_0, v)$ gilt:

$$\begin{aligned} p \equiv_A q &\Leftrightarrow \forall w \in \Sigma^*. \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F \\ &\Leftrightarrow \forall w \in \Sigma^*. \hat{\delta}(q_0, uw) \in F \Leftrightarrow \hat{\delta}(q_0, vw) \in F \\ &\Leftrightarrow \forall w \in \Sigma^*. \underline{uw} \in L(A) \Leftrightarrow vw \in L(A) \end{aligned}$$

Kongruenz

Beobachtung

Für $p := \hat{\delta}(q_0, u)$ und $q := \hat{\delta}(q_0, v)$ gilt:

$$\begin{aligned} p \equiv_A q &\Leftrightarrow \forall w \in \Sigma^*. \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F \\ &\Leftrightarrow \forall w \in \Sigma^*. \hat{\delta}(q_0, uw) \in F \Leftrightarrow \hat{\delta}(q_0, vw) \in F \\ &\Leftrightarrow \forall w \in \Sigma^*. \underline{uw} \in L(A) \Leftrightarrow vw \in L(A) \end{aligned}$$

Definition 2.58

Jede Sprache $L \subseteq \Sigma^*$ induziert eine Äquivalenzrelation

$$\equiv_L \subseteq \Sigma^* \times \Sigma^*:$$

Kongruenz

Beobachtung

Für $p := \hat{\delta}(q_0, u)$ und $q := \hat{\delta}(q_0, v)$ gilt:

$$\begin{aligned} p \equiv_A q &\Leftrightarrow \forall w \in \Sigma^*. \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F \\ &\Leftrightarrow \forall w \in \Sigma^*. \hat{\delta}(q_0, uw) \in F \Leftrightarrow \hat{\delta}(q_0, vw) \in F \\ &\Leftrightarrow \forall w \in \Sigma^*. \underline{uw} \in L(A) \Leftrightarrow vw \in L(A) \end{aligned}$$

Definition 2.58

Jede Sprache $L \subseteq \Sigma^*$ induziert eine Äquivalenzrelation

$$\equiv_L \subseteq \Sigma^* \times \Sigma^*:$$

$$u \equiv_L v \Leftrightarrow \forall w \in \Sigma^*. uw \in L \Leftrightarrow vw \in L$$

Kongruenz

Beobachtung

Für $p := \hat{\delta}(q_0, u)$ und $q := \hat{\delta}(q_0, v)$ gilt:

$$\begin{aligned} p \equiv_A q &\Leftrightarrow \forall w \in \Sigma^*. \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F \\ &\Leftrightarrow \forall w \in \Sigma^*. \hat{\delta}(q_0, uw) \in F \Leftrightarrow \hat{\delta}(q_0, vw) \in F \\ &\Leftrightarrow \forall w \in \Sigma^*. \underline{uw} \in L(A) \Leftrightarrow vw \in L(A) \end{aligned}$$

Definition 2.58

Jede Sprache $L \subseteq \Sigma^*$ induziert eine Äquivalenzrelation

$$\equiv_L \subseteq \Sigma^* \times \Sigma^*:$$

$$u \equiv_L v \Leftrightarrow \forall w \in \Sigma^*. uw \in L \Leftrightarrow vw \in L$$

D.h. u und v sind durch Anhängen von Wörtern bzgl. L nicht unterscheidbar.

Kongruenz

Beobachtung

Für $p := \hat{\delta}(q_0, u)$ und $q := \hat{\delta}(q_0, v)$ gilt:

$$\begin{aligned} p \equiv_A q &\Leftrightarrow \forall w \in \Sigma^*. \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F \\ &\Leftrightarrow \forall w \in \Sigma^*. \hat{\delta}(q_0, uw) \in F \Leftrightarrow \hat{\delta}(q_0, vw) \in F \\ &\Leftrightarrow \forall w \in \Sigma^*. \underline{uw} \in L(A) \Leftrightarrow vw \in L(A) \end{aligned}$$

Definition 2.58

Jede Sprache $L \subseteq \Sigma^*$ induziert eine Äquivalenzrelation

$\equiv_L \subseteq \Sigma^* \times \Sigma^*$:

$$u \equiv_L v \Leftrightarrow \forall w \in \Sigma^*. uw \in L \Leftrightarrow vw \in L$$

D.h. u und v sind durch Anhängen von Wörtern bzgl. L nicht unterscheidbar.

Obige Beobachtung lässt sich nun schreiben als

$$u \equiv_{L(A)} v \Leftrightarrow \hat{\delta}(q_0, u) \equiv_A \hat{\delta}(q_0, v)$$

Achtung

$p \equiv_A q$ ist eine Relation auf Zuständen von A

$u \equiv_L v$ ist eine Relation auf Wörtern

Kongruenz

$$u \equiv_{L(A)} v \Leftrightarrow \hat{\delta}(q_0, u) \equiv_A \hat{\delta}(q_0, v)$$

Da alle Zustände von q_0 erreichbar sind, gilt sogar: Die Abbildung

$$[u]_{\equiv_{L(A)}} \mapsto [\hat{\delta}(q_0, u)]_{\equiv_A}$$

ist eine Bijektion zwischen den $\equiv_{L(A)}$ und \equiv_A Äquivalenzklassen.

Kongruenz

$$u \equiv_{L(A)} v \Leftrightarrow \hat{\delta}(q_0, u) \equiv_A \hat{\delta}(q_0, v)$$

Da alle Zustände von q_0 erreichbar sind, gilt sogar: Die Abbildung

$$[u]_{\equiv_{L(A)}} \mapsto [\hat{\delta}(q_0, u)]_{\equiv_A}$$

ist eine Bijektion zwischen den $\equiv_{L(A)}$ und \equiv_A Äquivalenzklassen.

Satz 2.59

Ist A ein DFA ohne unerreichbare Zustände, so ist der von Algorithmus U berechnete Quotientenautomat A/\equiv ein minimaler DFA für $L(A)$.

Kongruenz

$$u \equiv_{L(A)} v \Leftrightarrow \hat{\delta}(q_0, u) \equiv_A \hat{\delta}(q_0, v)$$

Da alle Zustände von q_0 erreichbar sind, gilt sogar: Die Abbildung

$$[u]_{\equiv_{L(A)}} \mapsto [\hat{\delta}(q_0, u)]_{\equiv_A}$$

ist eine Bijektion zwischen den $\equiv_{L(A)}$ und \equiv_A Äquivalenzklassen.

Satz 2.59

Ist A ein DFA ohne unerreichbare Zustände, so ist der von Algorithmus U berechnete Quotientenautomat A/\equiv ein minimaler DFA für $L(A)$.

Beweis:

Sei $L := L(A)$ und A' ein DFA mit $L(A') = L$. Dann gilt:

$$|Q'| \geq |Q'/\equiv_{A'}| = |\Sigma^*/\equiv_L| \text{ Kongruenz}$$

Es gilt sogar (Übung!):

Fakt 2.60

Alle Quotientenautomaten A/\equiv_A für die gleiche Sprache $L(A)$ haben die gleiche Struktur, d.h. sie unterscheiden sich nur durch eine Umbenennung der Zustände.

Kongruenz

Es gilt sogar (Übung!):

Fakt 2.60

Alle Quotientenautomaten A/\equiv_A für die gleiche Sprache $L(A)$ haben die gleiche Struktur, d.h. sie unterscheiden sich nur durch eine Umbenennung der Zustände.

Daher beschriften wir die Zustände des kanonischen Minimalautomaten für eine Sprache L mit \equiv_L Äquivalenzklassen.

Beispiel 2.61

Sei $L := \{w \in \{0, 1\}^* \mid w \text{ endet mit } 00\}$.

Die einzigen drei \equiv_L Äquivalenzklassen sind:

$$[\epsilon]_{\equiv_L} = \{w \mid w \text{ endet nicht mit } 0\}$$

$$[0]_{\equiv_L} = \{w \mid w \text{ endet mit } 0, \text{ aber nicht mit } 00\}$$

Kongruenz

Es gilt sogar (Übung!):

Fakt 2.60

Alle Quotientenautomaten A/\equiv_A für die gleiche Sprache $L(A)$ haben die gleiche Struktur, d.h. sie unterscheiden sich nur durch eine Umbenennung der Zustände.

Daher beschriften wir die Zustände des kanonischen Minimalautomaten für eine Sprache L mit \equiv_L Äquivalenzklassen.

Beispiel 2.61

Sei $L := \{w \in \{0, 1\}^* \mid w \text{ endet mit } 00\}$.

Die einzigen drei \equiv_L Äquivalenzklassen sind:

$$[\epsilon]_{\equiv_L} = \{w \mid w \text{ endet nicht mit } 0\}$$

$$[0]_{\equiv_L} = \{w \mid w \text{ endet mit } 0, \text{ aber nicht mit } 00\}$$

$$[00]_{\equiv_L} = \{w \mid w \text{ endet mit } 00\} \text{ Kongruenz}$$